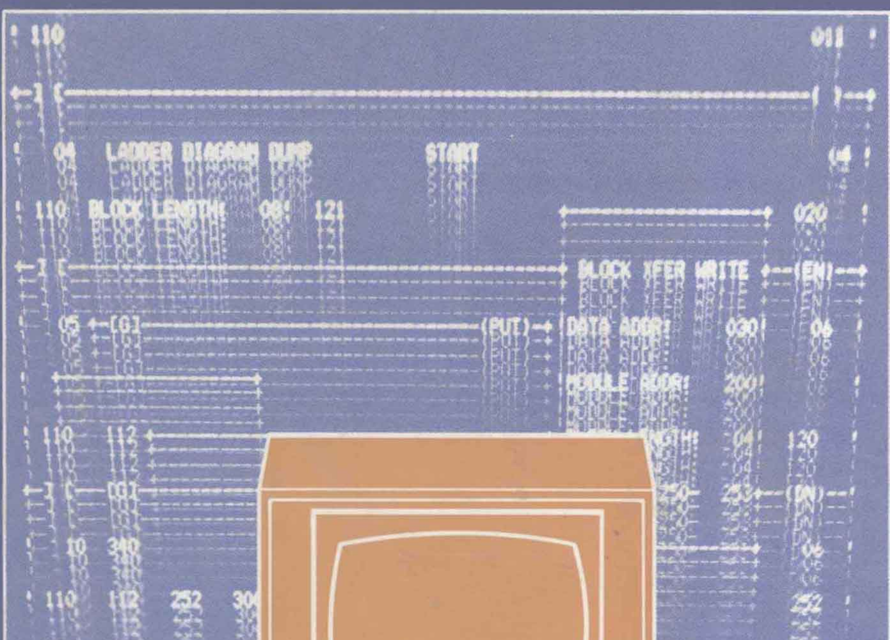
UNDERSTANDING AND USING Programmable Controllers



Thomas E. Kissell

Understanding and Using Programmable Controllers

Library of Congress Cataloging-in-Publication Data
KISSELL, THOMAS E.
Understanding and using programmable controllers.
Includes index.
1. Programmable controllers. I. Title.
TJ223.P76K57 1986 629.8'95 85-19095
ISBN 0-13-937129-X

Editorial/production supervision
and interior design: **Kathryn Pavelec**
Cover design: **Whitman Studio, Inc.**
Manufacturing buyer: **Rhett Conklin**

Printed in the United States of America

10 9 8 7 6 5 4

ISBN 0-13-937129-X 01

PRENTICE-HALL INTERNATIONAL (UK) LIMITED, *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
PRENTICE-HALL CANADA INC., *Toronto*
PRENTICE-HALL HISPANOAMERICANA, S.A., *Mexico*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*
EDITORA PRENTICE-HALL DO BRASIL, LTDA., *Rio de Janeiro*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

Understanding and Using Programmable Controllers

To my wife Kathleen
and my children Kelly and Christopher

Preface

This text is a comprehensive view of programmable controllers. It starts at an introductory level with the assumption that the reader has no previous experience on P.C.'s. Any knowledge the reader has about motor controls and digital electronics would be helpful, but is not necessary.

The author has carefully chosen programming examples that will reflect formats of the most widely used P.C.'s, since a format for every P.C. in use today cannot be presented in one book. It is the author's intention that the programming methods presented will provide the reader with the information needed to program similar types of P.C.'s.

Before reading this text, it should be noted that it was not written to replace the technical data that is supplied by P.C. manufacturers. Rather, this text should give the reader an essential understanding of basic P.C. theory and operation so that the manufactured material can be fully utilized.

The first four chapters of this text cover basic information concerning all P.C.'s. This information includes basic P.C. operation, history, numbering systems, and programming panels.

The next five chapters include programming examples to cover typical P.C. operations such as coils and contacts, timers, counters, sequencers, and math functions. These chapters explain how the major P.C. brands are formatted and programmed.

The next seven chapters give the reader an understanding of installation, troubleshooting, maintenance, and the operation of hardware like I/O modules and power supplies.

The last four chapters discuss advanced uses of P.C.'s for such things as acquisition and process control. Also included in this section are chapters on program documentation and advanced I/O modules.

Each chapter includes diagrams and pictures to help the reader understand P.C.'s. All chapters have questions at the end, and in some cases, programming exercises as well. The author hopes that the reader will try some or all of the programming examples to enhance his or her understanding of programmable controllers.

The author wishes to acknowledge the following people who supplied pictures and product information: Mr. Howard Hendricks from Gould Inc. Programmable Controller Division; Mr. Christopher Roy from Allen-Bradley; Ms. Phyllis Barr Fox from Texas Instruments; and Mr. Michael Waletzko from Square D Company.

I would also like to thank my wife Kathleen for all the work she put into typing this manuscript; and my children Christopher and Kelly, for their help in assembling parts of this material.

Thomas E. Kissell

Understanding and Using Programmable Controllers

Contents

chapter 1
Introduction

Programmable controllers, or P.C.'s, have been used in industry in one form or another for the past fifteen years. The newest models may not resemble their earlier counterparts, but many of the concepts used in the first units are still in use. The picture in Fig. 1-1 shows a typical programmable controller. The programmable controller is basically a computer-controlled system containing a microprocessor that is programmed with a programming panel, or keyboard. The P.C. is dedicated to receiving input signals and sending output signals in response to the program logic. The program generally consists of contacts, outputs, timers, counters, and math functions. The programmable controllers found in industry today have evolved from the need for a control system that can be easily reprogrammed as changes occur or as new products develop.

For example, the automotive industry is faced annually with major changes in production as new models are designed. This changeover previously required electricians and maintenance personnel to put in long hours to rewire relay-type controls. Each changeover period was costly to the industry, and it often forced changes to be infrequent and as simple as possible.

The programmable controller was developed in 1969 to ease the problem of changing control systems periodically. Modern P.C.'s consist of four major parts: a central processing unit, also known as the processor or computer; an input section; an output section; and a power supply (see Fig. 1-2). The programming panel is not considered a major part of an operating P.C. because once you have used it to program the processor, it can be disconnected and the system will still operate correctly. The relationship of these four parts and the ability of the P.C.'s to be

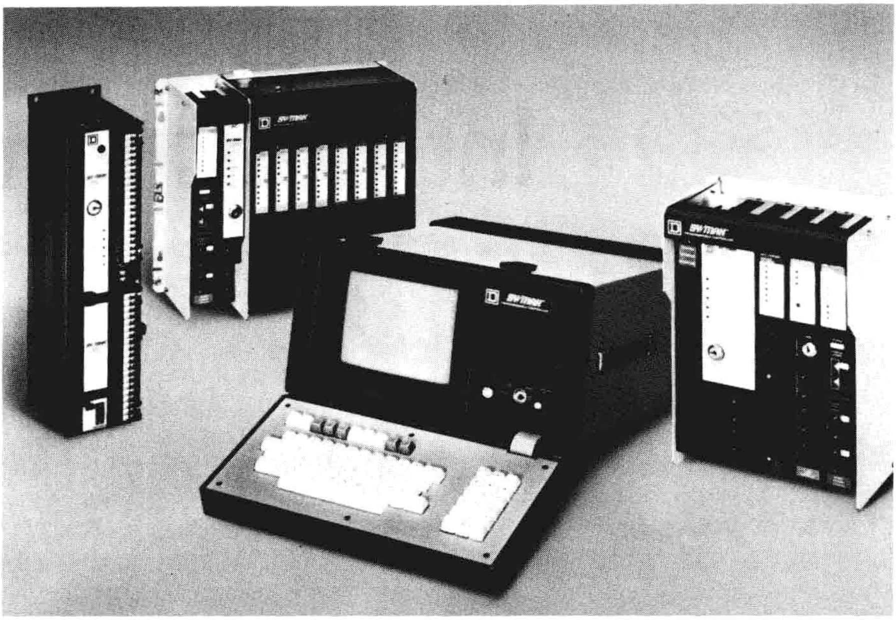


Figure 1-1 A typical P.C. system. Picture courtesy of Square D Company.

reprogrammed can best be understood by comparing hardwired circuits to identical circuits controlled by a P.C.

From Fig. 1-3 you can see that Switch 1 and Switch 2 are normally opened push-button switches. Switch 1 will send power to Lamp 1, and Switch 2 will send power to Lamp 2. When Switch 1 is pushed closed, Lamp 1 will light. When Switch 2 is closed, Lamp 2 will light.

Figure 1-4 shows the same components connected to a P.C. From this diagram you can see several differences. First, switches are not connected directly to the lamps, instead, the *switches* are connected to *input modules*, and the *lamps* are connected to *output modules*. Another difference is that the input modules and

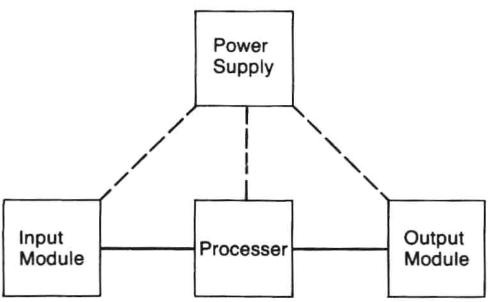


Figure 1-2 Block diagram of a P.C.

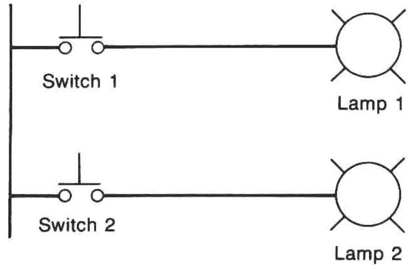


Figure 1-3 Hardwired system.

output modules are not connected to each other directly. They are connected through the *processor* when the program logic indicates certain conditions have been met.

The processor in Fig. 1-4 is programmed to connect Switch 1 to Lamp 1, and Switch 2 to Lamp 2. This is accomplished by typing a program or diagram into the processor from a keyboard. The program looks very similar to a regular electrical ladder diagram.

The operation of the hardwired switches and lamps and the P.C.-controlled system seem identical. When Switch 1 closes, Lamp 1 lights, and when Switch 2 closes, Lamp 2 lights. The major difference is in the way electricity flows to accomplish this.

In the hardwired system, the electrons flow from the voltage source, through the switch, to the correct indicator lamp. Electrical power simply follows the wire conductors to the lamp. When the switch is opened, power is interrupted and the light goes out.

In the P.C.-controlled system, electrical power comes from the voltage source, through the switch, into the input module. The input module senses the presence of this voltage and in turn, sends a small signal voltage into the processor. The voltage from the switch is isolated from the voltage signal that the module sends

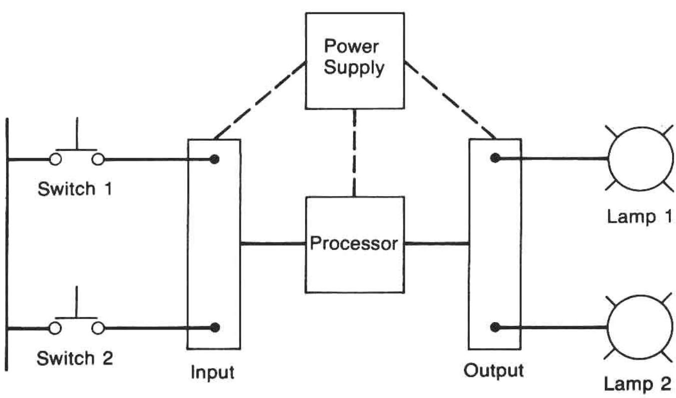


Figure 1-4 P.C. system.

into the processor. This isolation is necessary since the fragile processor chip operates at very low voltage and current levels. Isolation is generally provided by an electronic component known as an *opto coupler*, which enables the power supply to provide the appropriate voltage levels to input modules, output modules, and the processor.

The processor receives a signal from the input module when the switch is closed, and will send a similar signal to the output module as directed by the program. The program directs the processor-transmitted signal to the appropriate output terminal for Lamp 1 when it receives a signal from the input module terminal connected to Switch 1. All of this takes place in less than a thousandth of a second.

When Switch 2 is activated a similar action is completed by the processor, but this time the signal is sent to the output module terminal for Lamp 2.

An observer of both hardwired and P.C.-controlled systems would not notice any difference in system operation. In both systems, Switch 1 controls Lamp 1, and Switch 2 controls Lamp 2.

The biggest advantage of the programmable controller becomes evident when a change is needed in the circuits previously discussed. For example, if you needed to change the circuits of a hardwired system to have Switch 1 control Lamp 2, and Switch 2 control Lamp 1, it would take nearly ten minutes to rewire them, and would involve exchanging the wires at the switches or the lamps.

The same change in a P.C.-controlled system is accomplished by reprogramming the P.C. Since changes can be made in the program instead of the wiring system, many man-hours are saved.

This simple example of changing a program can be applied to a system containing fifty or more switches and outputs. You can see that by merely editing or changing the program, a great deal of time can be saved.

Now that you can see the major advantage of using a programmable controller, we can proceed with more information about the basic system. In the previous example, the P.C. used a program to determine which lamp to turn on. This program is entered into the processor through the keyboard, stored in the processor's memory, and displayed on a screen called a *CRT* (cathode-ray tube). The CRT is similar to the tube used in televisions and allows the P.C. programmer to view the present diagram that is in the processor's memory. The program on the CRT screen looks very similar to a regular electrical diagram and uses normal electrical diagram symbols. Another feature of most P.C.'s is their capability of being connected to a printer, so that a hard-copy printout can be made of the electrical diagram. The hard-copy printout of the program is used for troubleshooting.

Other usable features on most programmable controllers include the availability of timers and counters with programmable times and counts. These timers and counters can be programmed to simulate the operation of electromechanical timers and counters. The maximum number of counters and timers in a program depends on the size of processor's memory. The processor chip also gives P.C.'s the normal features of other computers, such as addition, subtraction, multiplication, and division functions.

MORE ABOUT PROGRAMMABLE CONTROLLERS' BASIC PARTS

Figure 1-4 shows the input and output sections of the programmable controller as rectangular boxes. The boxes are used to show simplicity. Figure 1-5 shows a better view of the input and output sections of a P.C. You can see that the input and output modules are connected side by side in the I/O module housing. At first, all you need to know about input and output modules is that they are needed to receive signals from switches and send signals to output coils and lamps, as directed by the processor. In later chapters you will become more familiar with their internal operation and how the processor identifies them.

You will also find that even though most P.C. input, output, and specialty modules all look alike, they have a variety of functions. They are also wired in a variety of ways. Some differences will even be apparent between various models made by the same manufacturer.

The input and output section is used to interface the P.C. with many industrial operations and machines. This means each input and output listed in the program must be given a number for identification. This number is also used as a storage location for the program device inside the processor memory and as an address location for the input or output module in the housing. At first, this may seem a little confusing, but you will soon understand that every component in the program must be numbered so you can identify it. The processor uses the address to send and receive signals to the modules. The processor must also use the identification number to remember where data pertaining to the programmed device is stored in memory. Finally, you will use the identification number as the address to physically

Figure 1-5 Picture of I/O modules and processor. Picture courtesy of Gould Modicon Programmable Controller Division.

locate the input and output modules in order to wire and troubleshoot them. Chapter 5 discusses this numbering system in depth. Until then you will only need to be aware that the numbering system exists and that it is vital for the P.C. system to operate correctly. You will also learn that each manufacturer designs its own numbering system. All these systems are somewhat similar and easy to learn.

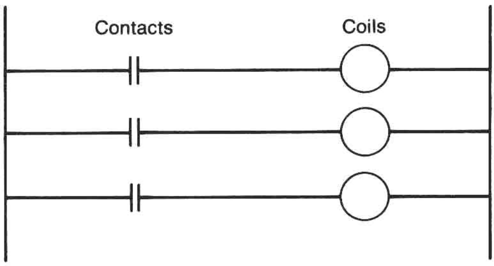


Figure 1-6 Ladder diagram format.

AN EXAMPLE PROGRAM FOR PROGRAMMABLE CONTROLLERS

The basic program format for programmable controllers is very similar to an electrical ladder diagram. The ladder diagram is so-called because the lines of a completed diagram resemble the rung of a wooden ladder (see Fig. 1-6). In fact, some P.C. manufacturers like Allen-Bradley refer to each program line as a rung.

The ladder diagram is constructed to show the sequence of events, rather than the path of wires and the layout of the electrical parts. Figure 1-7 shows a typical start-stop circuit controlling a motor starter. The circuit is drawn as a wiring diagram that shows component placement and wiring paths in Fig. 1-7 and as a ladder diagram that shows the sequence of operation in Fig. 1-8.

You should notice that the wires in the wiring diagram may cross one another and become hard to follow, even though the circuit has only three components. This happens because accurate location of components is the main function of the

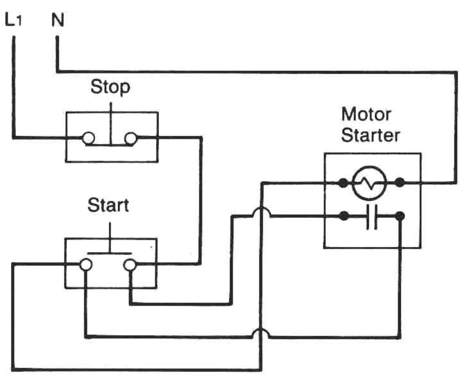


Figure 1-7 Wiring diagram of start-stop circuit.

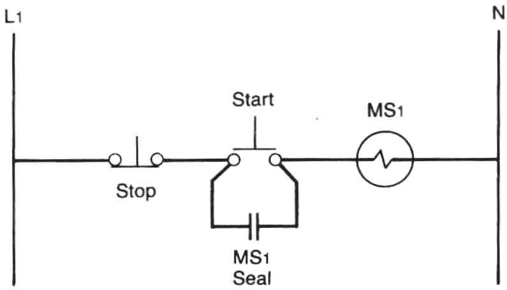


Figure 1-8 Ladder diagram of start-stop circuit.

wiring diagram. You can imagine how many wires might cross if the diagram was showing fifty components.

In comparison, you should notice that in the ladder diagram the wires should never cross. Power is provided to each rung of the diagram from the vertical line that runs down the left side of the diagram. A vertical line also runs down the right side of the diagram to provide power to complete the circuit. These two vertical lines are usually called L1 and L2 for line 1 and line 2, or L1 and N for line 1 and neutral. These are the terminals where power is provided from the transformer. Each line, or rung, generally has one coil, or load, and can have as many switches or contacts as is required to control the load. Some P.C.'s will limit the total number of contacts and coils in each program line or rung.

Another important difference that you should notice when looking at the wiring diagram in Fig. 1-7 and the ladder diagram in Fig. 1-8 is that the coil of the relay, or motor starter, is shown very near the contacts it is controlling. This is because the coil needs to be physically very close to its contacts so that its magnetic force can open or close its contacts. Remember the wiring diagram is drawn to show the true location of each part, or component. In comparison, the coil in a ladder diagram may show up several rungs or even pages away from the contacts it controls because it sequentially occurs this way. In other words, the contacts of a relay or motor starter may show up anywhere in the ladder diagram. This may seem difficult to follow at first, but each contact is identified by a number or name to tell which coil operates it. All you need to remember is that the coil must be energized before its magnetic field can move the contacts. It may be easier if you make a note right above or near each contact set you find in a program, indicating the name of the relay coil that controls it. That way you won't become confused by finding contacts of one relay in several different program lines.

The typical programmable controller program looks very similar to a ladder diagram. This format is used because the ladder diagram has been the working language of electricians for many years. This format style is also used because the computer scans the program in a sequential manner. Some early programmable controllers tried to use BASIC, Pascal, and FORTRAN computer languages to control inputs and outputs. Even though these were commonly used computer languages, they did not succeed as languages for programmable controllers because electricians could not understand them to troubleshoot and repair large industrial

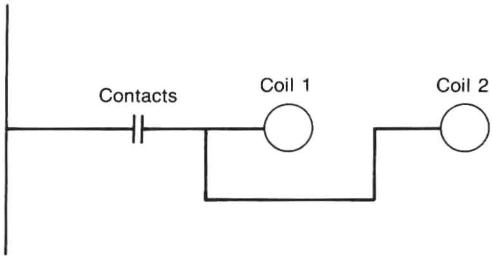


Figure 1-9 Two coils in series.

systems. Since programmable controllers now use the ladder diagram-type program, they have truly become an electrician's control system.

The actual programming formats for each programmable controller may be a little different from each other, but they all resemble the original electrical ladder diagram. Most systems still refer to the vertical line running down the left side of the program as the source for all power to each program line or rung. Some manufacturers even refer to this line as the power rail. For this reason, when you connect contacts and coils in a circuit, the beginning of that circuit must touch the left side of the program to get power. The format for some P.C.'s requires you to start each program line at the power rail and then work to the right side. They also require that you put at least one load in each circuit. This load will usually be a coil, but you could use timers, counters, or other similar function blocks for the load. Some brands of P.C.'s allow you to put several loads on the same line. This appears to put the loads in series. All electricians know that loads, such as coils, will not function correctly if they are connected in series. What has really happened in the P.C. is that solid state electronic chips accommodate this in the processor's program, but the coils or outputs actually operate as though they are in parallel. In other words, if two coils are in series in the program as shown in Fig. 1-9, they operate as though they are in parallel.

If the first coil does not operate for some reason, the second one will still function. This is one major difference between the electronic outputs in a programmable controller, and electromechanical devices, such as coils and solenoids, in a hardwired system. As you complete this text, you will notice several other differences like this that allow you to do time-saving functions on the programmable controller. Remember, some P.C.'s will not allow two coils or loads on the same program line, so be sure to check the format of your P.C.

Another programming feature that tends to bother someone trying to learn the programmable controller for the first time is the method that P.C. manufacturers use to explain how their program is scanned in memory. This may be accomplished by the processor starting at the top left side and scanning each column from top to bottom, then moving to the next column and scanning it from top to bottom. Other processors may scan rung by rung. Remember the computer chip, or processor, has its own method for entering, storing, and executing a program. You will only need to know the general operation of the processor, since some specific instructions will not be functional due to the type of scan.